DOCUMENT RESUME

ED 301 185 IR 013 525

AUTHOR Linn, Marcia C.
TITLE Science Education and the Challenge of Technology.
PUB DATE Apr 88
NOTE 39p.; Paper presented under the title "Technology and Instruction in the Sciences" at the Annual Meeting of the American Educational Research Association (New Orleans, LA, April 5-9, 1988).
PUB TYPE Information Analyses (070) -- Viewpoints (120) -- Speeches/Conference Papers (150)

EDRS PRICE MF01/PC02 Plus Postage.
DESCRIPTORS Courseware; *Curriculum Development; Databases; *Educational Research; *Educational Technology; Elementary Secondary Education; Expert Systems; Higher Education; Instructional Innovation; *Laboratory Experiments; Programing; *Science Education; *Technological Advancement
IDENTIFIERS *Thinking Skills

ABSTRACT

This analysis of the relationship between science education and technology over the past 15 years focuses on lessons learned from research on learning and instruction, the relationship between the teacher and technology, and the influence of technological advances on educational practice. Five conclusions are drawn: (1) courseware authoring environments that facilitate the reformulation of educational software will improve science instruction; (2) taking full advantage of the talents of teachers in modeling scientific processes, encouraging students to learn, and helping learners gain coherent understanding will improve science education; (3) substantial trial and refinement of innovations in realistic settings will help students reach their potential; (4) the goals of science education will need reformulation to help impart the thinking and reasoning skills that students need in the information age; and (5) effective technological innovation in science education depends on true collaboration among curriculum developers, precollege teachers, educational researchers, school administrators, and state and national policymakers. The text is supplemented by four figures. (73 references) (EW)

****
* Reproductions supplied by EDRS are the best that can be made *
* from the original document. *
****

ED3011185

U.S. DEPARTMENT OF EDUCATION
Office of Educational Research and Improvement
EDUCATIONAL RESOURCES INFORMATION CENTER (ERIC)
☒ This document has been reproduced as received from the person or organization originating it.
☐ Minor changes have been made to improve reproduction quality.
• Points of view or opinions stated in this document do not necessarily represent official OERI position or policy.

Science Education and the Challenge of Technology[1]

"PERMISSION TO REPRODUCE THIS MATERIAL HAS BEEN GRANTED BY
Marcia C. Linn
TO THE EDUCATIONAL RESOURCES INFORMATION CENTER (ERIC)."

Marcia C. Linn
Graduate School of Education
University of California, Berkeley
Berkeley, California 94720

This paper was presented in a symposium for the
American Educational Research Association
New Orleans, Louisiana, April, 1988
under the title:
"Technology and Instruction in the Sciences"

Symposium Title:
Classroom Learning and the Challenge of Technology
Symposium Organizer:
Marcia C. Linn

[1]Chapter to appear in the **1988 Yearbook of the Association for the Education of Teachers in Science** (AETS), entitled *Informational Technologies and Science Education*. Published by ERIC Clearinghouse for Science, Math, and Environmental Education.

This material is based upon research supported by the National Science Foundation under grant nos. DPE-84-70364 and MDR-84-70514. Any opinions, findings, and conclusions or recommendations expressed in this publication are those of the author and do not necessarily reflect the views of the National Science Foundation.

The author gratefully acknowledges the contributions made by the students and teachers of the Autonomous Computer Classroom Environments for Learning (ACCEL) Project and the Computer as Lab Partner (CLP) Project.

IR013525

Introduction

In this chapter, I analyze the relationship between science education and technology over the last 15 years, identifying promising trends, and recommending policies for the future. I focus on (a) lessons learned from research on learning and instruction, (b) the relationship between the teacher and technology, and (c) the influence of technological advance on educational practice. The potential impact of technology on science education comes at a time when students, teachers, policy makers, and numerous comparative assessments suggest than science education in America requires a major overhaul.

Why might technological advance catalyze improvement in science education? First, scientists use constantly advancing technology in laboratories and offices to help solve complex problems. These tools of experts might also help students. Second, recent technologies have already started school. Currently, precollege students have access to over 1.4 million computers at school and more at home (e.g., Becker, 1986). Third, the information explosion changes the skills students need, and electronic databases will make this information available in schools. Fourth, technology has transformed the workplace by taking over manufacturing and other functions. As a result, current students will probably change jobs several times during their careers and therefore need skills in learning new information. Fifth, educators use rapidly advancing technological tools. Such tools can help teachers simplify tedious record-keeping and secretarial tasks, gain access to colleagues and information, and enhance instruction. Sixth, recent research on learning clarifies how scientists solve complex problems. These findings suggest how technological tools could be used to teach problem solving skills to students. Thus, technological advance can improve science education and science educators have the opportunity to respond to current shortcomings of science instruction by harnessing technology.

The Current State of Science Education

At present, there is widespread agreement that science education is not effective for American students. A recent comparison study revealed that both average and the very best American students performed well below their counterparts from most other industrialized countries on test

items reflective of the current science curriculum (see Figures 1 and 2, National Science Foundation [NSF], 1987). This survey reinforces findings from many other national and international assessments (International Association for the Evaluation of Educational Achievement, 1988; National Assessment of Educational Progress, 1978; National Commission on Excellence in Education, 1983). Furthermore, the percentage of foreign students enrolled in graduate science programs in American universities has increased from 21% in 1980 to 28% in 1986 (NSF, 1987), and the number of American students has stabilized. Other indicators also suggest that American students are not maintaining their past standards. For example, a comparison between the quality of patented ideas in the United States and Japan reported a tentative trend towards an increase in the number of high quality patented ideas in Japan compared to the United States (Broad, 1988; NSF, 1987). In summary, performance on national standardized tests, enrollment in advanced scientific programs of study and analysis of the quality of patented ideas all suggest that the United States is losing its edge in scientific accomplishment. A new, improved approach to science education is needed.

Although standardized test scores signify serious problems with American science education, they do not dictate a solution. Indeed, they often trigger increasing drill and practice, which probably contributes to poor performance, because information that is not cohesively integrated is rapidly forgotten (Linn, 1987a). The response of increasing drill is reinforced by science textbooks that (a) emphasize science information rather than scientific reasoning and (b) cover topics so quickly that students cannot integrate information or apply what they learn to new problems. Emphasis on memorization deters students from integrating and understanding scientific phenomena and detracts from problem solving and self-monitoring. Instead, given rapid technological advance and the information explosion, students need skill in solving new problems, not skill in memorizing scientific information which can be readily accessed in electronic databases or which will be outdated by the time they graduate.

To achieve the robust, cohesive understanding of science, students need a realistic, philosophically-sound view of science. Rather than memorizing and absorbing information,

students should expect to analyze and question information, to synthesize and integrate their knowledge, and to cooperatively solve problems that require understanding of scientific principles. Students must be expected to apply information learned in one situation to similar problems in other situations. With these skills, students will perform better on standardized tests as well as on indicators of problem-solving skill. Furthermore, students will understand that the business of science is discovering new information.

A minority of science programs do emphasize a philosophically-sound view of science using "discovery learning," "hands-on" experimentation, or scientific problem solving to illustrate the process of science. Although promising, these approaches rarely go far enough. Often discovery learning experiences are unguided, so students have no idea how to proceed, or conducted without the details students need to draw conclusions. Unless these experiences are integrated and systematic, students end up with fragmented and incomplete understanding. To be effective, discovery learning takes instructional time and teacher expertise. Knowledge will not be integrated if, just when students get started on a really interesting problem, they need to stop because the fifty-minute class period has elapsed or the week devoted to thermodynamics is over. Students learn complex problem-solving skills when individually guided by knowledgeable teachers. For example, Bloom (1984) concludes that tutoring might be two standard deviations more effective than traditional instruction. Yet those who currently teach science often lack the science knowledge and pedagogical skills required to tutor students on complex problems (National Science Board, 1988).

In summary, although a philosophically-sound view of science education has been difficult to implement, drill on science information is not the alternative. Many believe technological advances can help implement discovery activities, can free teachers to spend more time tutoring individuals and small groups of students, and can help prepare teachers for this role. Several projects, discussed subsequently, that use technology in innovative science programs point the way.

In this paper, I argue that we can achieve the kind of science education American students deserve by: (a) combining recent technological developments with recent advances in

understanding how students learn and what makes instruction successful, (b) establishing collaborations between curriculum developers, educational researchers, teachers, and policy makers, and (c) refining preliminary approaches through trials in realistic settings.

Integrating Technology into Science Education

To reach the current use of technology, science education has passed through three discernible stages. In this developmental process, those involved are jointly constructing an understanding of how technology and science education can complement each other. Factors influenced by the process include the goals of science education, the role of the teacher, and the nature of technology..

Stage I: Technology in the Service of Established Goals

At first, developers targeted technological innovations to the established goals of science education rather than seeking creative uses for this new medium. Developers were often isolated, unaware of the efforts of others, unaware of educational research that might influence their efforts, and optimistic about the ease with which technological tools could improve science education. For example, precollege science teachers and their students wrote multiple-choice quizzes and created question and answer type software. Even these traditional goals were difficult to achieve because developers lacked tools for including graphics or building robust interfaces between the computer and the student, and therefore often failed to use technology effectively, as the following examples illustrate.

Imitating Textbooks

Early science software imitated textbooks by placing paragraphs on the computer screen and interspersing questions. As a result, most of the disadvantages of texts, but few of the advantages, were achieved. In particular, (a) students and teachers have skills for browsing through textbooks that do not apply to browsing through screen-presented text, (b) comprehension falls when text appears on a computer screen, and (c) access to computer-presented text is, of necessity, constrained by the availability of the hardware, limiting access to school hours.

Drill and Practice

Many developers used computers for drill and practice, consistent with the goal of memorizing scientific information. This approach draws on principles from Skinnerian learning theory, on user interfaces designed for electronic games, and on motivating students by keeping score. For example, a chemistry program designed to help students memorize how compounds interact involves (a) skill in directing the cursor around a computer-presented maze while being chased, (b) rapid decision-making concerning which compounds will interact with which other compounds, and (c) strategic thinking about the game-like scoring system. The emphasis on eye-hand coordination excites students, but may overwhelm the scientific content. In addition, this software emphasizes absorption of isolated pieces of information to get a higher score, rather than encouraging students to understand the material.

Simulating Science Experiments

Teachers often omit experiments from the curriculum for lack of equipment, inaccessibility in the classroom, lack of instructional time, potential for danger, or potential for disrupting learning. As a result, software developers have made experiments available on computers. These computer-presented experiments are designed to (a) provide access to experiments, (b) emphasize some of the problems that arise in the lab such as selecting the correct instrument, zeroing the balance scale, placing samples properly, or interpreting results, or (c) extend experimentation to problems not amenable to classroom instruction.

Bork (1980) created computer-presented experiments to replace classroom experiments. In one example, students connected batteries and bulbs with wires to simulate electrical circuit experimentation. Bork argued that this approach freed the teacher from worrying about whether the batteries were charged and the bulbs were working, and allowed students to focus on observing differences in the intensity of the bulb depending on the number of batteries connected. Many science teachers disagreed, pointing out that experiments with batteries and bulbs were (a) cheap and easy to conduct in classrooms, and (b) one of a small set of experiments amenable to classroom instruction. The simulation was not fully tested because the hardware never achieved widespread use.

Tribbles (Von Blum & Hursh, 1987) provided access to unavailable experiments by simulating genetics experiments with fruit flies. The program allows students to analyze either the phenotypes or genotypes of the resulting population. Students can predict characteristics of offspring, set up experiments, and analyze the outcomes. Since the first version, more powerful hardware and software have permitted development of effective graphics and refinement of the interface. The revised program is a popular software tool today.

Widely-used software for science education was developed for a mainframe computer by the PLATO Project at the University of Illinois (Smith & Sherwood, 1976). With the advent of microcomputers, those developers used their considerable experience to develop microcomputer-based approaches to science experiments. The chemistry simulations developed by Stan Smith and Loretta Jones at the University of Illinois are one example. Using the IBM Info Windows touch screen, combined with a videodisc, this approach allows students to participate in completely simulated experiments. Students experience many of the same problems that might arise in a laboratory, such as selecting the correct instrument for their experiment, placing sample material in the instrument correctly, setting the background appropriately, and interpreting the result. The software features simulated experiments that would be difficult for students to conduct on their own due to the requirement of collecting data in remote locations, the need to use expensive equipment, or dangerous chemicals. Students make decisions as they would if they were experimenting on their own, such as where to place a collector to sample air quality and how to display and analyze the data. The advantage of the simulated experiments is that students' attention is directed to the main decisions that need to be made. On balance, students may become immersed in the details of conducting the experiment and lose track of the implications of their findings, just as they do when conducting experiments in real laboratories. The developers are analyzing the advantages of (a) using simulations by themselves, (b) using simulations for pre-lab instruction, and (c) using simulations for post-lab review.

Thus, efforts to simulate science experiments offer promise for science education. Successful programs have benefitted from trial and revision. Furthermore, many excellent efforts have been thwarted by changes or limitations in the hardware and software environments.

The Teacher Dialogue

A valued component of science instruction is the dialogue between teacher and student. Often teachers conduct dialogues with the class or with individuals to communicate important science concepts. The dialogues of Socrates or Rousseau's (1892) teaching of *Emile* serve as models for this approach.

Arons, Bork, Franklin, Kurtz, and Collea (1981) and others have taken the teacher dialogue as a model for the development of science software. They observe teachers using dialogues in classrooms and attempt to implement the practices of teachers in a computer program for students. Rather than using classroom apparatus, the program features illustrations and dynamic diagrams of scientific phenomena. Although some dialogues achieve what appears to be interactive discussion, these dialogues are limited by the ability of the computer to process natural language (e.g., Bork, 1981). Therefore, if students fail to use key words, the program acts more like a lecturer than an interactive companion. These dialogues have the property of encouraging students to reflect on problems and generate solutions. Difficulties arise when students come up with answers the computer cannot recognize. Then the process resembles an adventure game where the user seeks words the game will interpret. Furthermore, human tutors motivate students to continue to think about complex tasks by providing encouragement and empathy not possible with most dialogues (Lepper & Chabay, 1985). As a result, the cognitive effects of computer dialogues are likely to differ from those achieved by teachers.

Demonstrations Using Technology

Science classes often feature demonstrations. Teachers demonstrate experiments, illustrate relationships, and graph results. Many demonstrations can be presented by computer. For example, developers have created a simulation of the periodic table that allows teachers to dynamically illustrate the heating of the universe from absolute zero to a point where virtually every

element is vaporized. In another example, the developers of Rocky's Boots, a dynamic simulation using "and," "or," and "not" gates to construct "machines" can be used to illustrate how these gates perform. The simulation provides a dynamic trace of the activation pattern for the constructed circuit, going beyond what would be possible without technology (Stein & Linn, 1985).

Thus, software can help teachers demonstrate scientific phenomena in ways that would be difficult or impossible without technology. On balance, demonstrations, like lectures, often reach only a small proportion of the audience and may emphasize isolated phenomena, not integrated understanding.

Writing and Computation Aided by Technology

Most science classes involve writing and computation that could be performed using standard software for word processing and data analysis. Early experiences with these tools in the classroom suggest that they have some important advantages. Students now regularly learn to use scientific calculators with numerous functions and, as a result, focus more on problem interpretation than on tedious calculations.

How can word processing facilitate scientific report writing? Striley (1988) compared students writing collaborative reports using paper and pencil with those writing reports on the word processor. With paper and pencil, students often divided responsibility, saying, "You think and I'll write." In contrast, in the word processing option, students jointly constructed sentences and revised their reports. Using the word processor, students could refine their ideas and still produce neat reports. They were reluctant to erase and refine their ideas using paper and pencil. Thus, an unanticipated consequence of using wordprocessing was increased collaboration among students.

Summary

In summary, many software developers initially sought ways to use technology to implement practices already common in science classrooms, rather than move toward more philosophically-sound practices. However, programs emphasizing drill and practice on science information reinforced an already questionable goal for science instruction and demonstrations of scientific phenomena tended to perpetuate the lecture. In contrast, efforts to use technology for simulating

experiments and wordprocessors for writing reports revealed potential advantages of technology in science education. At this stage, the limitations of hardware and software caused real problems. Graphical representation of scientific phenomena required extensive programming, hardware installed in schools was abandoned by manufacturers, and many ideas could not be implemented given the power of school computers.

Efforts to use technological tools for existing goals in Stage I revealed uncertainty about the interaction between technology and the teacher. Some, such as the teacher dialogues, took over major functions performed by the teacher. Others, such as drill and practice or simulated experiments, take over functions performed by books or apparatus in the past, leaving the teacher to focus on integrating and synthesizing the information presented. These efforts paved the way for teachers and schools to rethink the role of the teacher in the classroom of the future.

Stage II: Adapting Science Education to Technological Innovation

In the history of science, technological innovation has shaped the direction of scientific advance on the one hand, and individuals have harnessed technological tools to redirect scientific investigation on the other. In the second stage of the relationship between technology and science education, technological tools reshaped science education and science educators stimulated technological innovation. Instead of using technology for existing goals, these tools were focused on problem solving and complex reasoning skills. Two factors predominated. First, developers made technological tools used by expert scientists available to students, arguing that tools that help experts solve problems could teach problem solving rather than sustaining questionable practices currently in classrooms. Second, developers collaborated with cognitive researchers and combined advances in understanding learning and instruction with technological advance to teach a more philosophically-sound view of science teaching.

As the examples below illustrate, since science students are not experts, the tools of experts did not automatically impart the problem-solving skills of experts. Rather, these tools provided an opportunity for teachers, researchers, and curriculum developers to focus on roles and materials that would help students develop complex reasoning skills.

As science educators incorporated these tools, they benefitted from more powerful software development environments that made revision easier. In addition, hardware developers who created upwardly-compatible advances were embraced by schools because the new hardware did not make the established software obsolete.

Programming

Programming was the first expert use of technology implemented for students. Many scientists solve problems by writing their own computer programs to analyze data and to display information collected in their laboratories. As a result, some argued that students in science classes would learn problem solving from instruction in programming.

Perhaps the most well-known argument for programming is found in Papert's *Mindstorms* (1980). Inspired by the developmental theory of Piaget, Papert argued that students using powerful programming environments would have "wonderful ideas" about science. He developed Logo to allow students to explore scientific phenomenon by writing programs. For example, using turtle graphics, students can give direction to either a screen turtle or a robot-like floor turtle and examine the response to these instructions.

Papert and others report exciting insights gained by students using Logo and turtle graphics. Papert describes students who examine the consequences of having the turtle move both east and north simultaneously and discover that the turtle moves along the resultant vector (Lawler, 1986; Hughes & Macleod, 1986; Turkle, 1984). Lawler (1986) reports designing Logo programs for his young children that result in powerful insights into mathematics and science. Investigations by many researchers, however, reveal that Logo's success is largely determined by the efforts of sensitive teachers.

Investigations in realistic settings reveal that Logo does not, by itself, inspire students to develop wonderful ideas about science. Proficiency in Logo takes considerable time and students rarely build tools to help with other problems (Pea & Sheingold, 1987). Students are not experts and do not automatically think like experts. Just as other discovery learning environments rarely

succeed, so Logo used without expert direction rarely yielded effective learning (Pea & Kurland, 1987).

Yet, environments such as Logo make it easier to build tools targeted to specific goals of instruction. Creative uses of Logo did result from the development of teaching tools. Lawler (1986) developed tutorials for his children rather than having his children learn Logo. Logo heralded an important trend. It became an effective courseware authoring tool, allowing developers to create exploratory environments and provide guided discovery on important concepts.

Findings for Logo are consistent with investigations of other programming languages such as BASIC and Pascal. Many researchers concluded that students have difficulty learning programming, that much time elapses before they got to serious problem solving, and that they needed instruction in the problem-solving skills of experts, not free exploration of a programming language. In particular, experts engage in design, draw on a repertoire of algorithms or templates, and reflect on the flaws in their reasoning. These skills are rarely taught (Linn, 1985; Mandinach & Linn, 1987; Sloane & Linn, in press).

Recently, programming environments have been refined to make the debugging tools of experts more available. For example, Macintosh Pascal, shown in Figure 3, offers many of the debugging tools used by experts. Experts tend to debug programs by tracing the values of variables while running a program. Macintosh Pascal allows the student to do a trace by requesting it from a menu. A moving finger indicates which line of the program is being evaluated, an observe window indicates the values of the variables at the time that that line is being executed, and an output window indicates the state of the output, which in this case is graphic, at the same time. Paralleling findings for Logo, much research reveals that simply having this tool available does not help students develop the reasoning skills required for using it (Nachmias, Friedler, & Linn, 1988).

Thus, precollege programming courses rarely achieve the knowledge level that permits use of problem-solving skills. Furthermore, even if the syntax is learned, students need instruction in the thinking skills used by experts.

Microworlds

To encourage students to use the thinking skills experts use to investigate scientific problems, developers have created environments called "microworlds" that provide feedback about some scientific phenomena. Microworlds allow students to develop hypotheses about complex scientific phenomena, to design experiments to test these ideas, and to use the feedback to reflect on their conception of the phenomena, just as experts develop models and simulations to test their ideas. Although the microworlds generally require much less learning of syntax than programming, they share with programming a need for instruction in generating ideas and using feedback from experiments in order to succeed.

For example, the dynaturtle (di Sessa, 1979, 1986) allows students to explore Newton's laws of motion but does not teach students how to use the information they acquire. Before coming to science class, students observing the world around them construct views about motion that in many ways contradict Newton's laws. They conclude that objects in motion tend to slow down. They argue that when you push an object it moves in the direction that you push it, rather than assuming that a new force adds to previous forces that have acted on an object. Furthermore, individuals invent forces to explain the behavior of objects (Reif, 1987; Clement, 1987). These conceptions of the natural world become quite robust and cohesive by the time students encounter physics instruction and tend to persist even after students have learned Newton's laws in science classes. Frequently, students conclude that physics learned in science class applies to problems encountered in science class but not to objects in the natural world.

Developers hypothesized that microworlds could provide robust and cohesive alternatives to students' misconceptions. A question, however, is whether students would integrate their microworld experiences with their other experiences or assume that the physics applied using microworlds does not apply in the natural world.

Further exploration with the dynaturtle microworld by White (1981, 1984) revealed that students had difficulty linking experiences with the dynaturtle to experiences with the natural world. As a result, White and Frederiksen (1987) have expanded the notion of a microworld into a progression of microworlds, each coming closer to experiences students encounter in the natural world. They used what they call the ThinkerTools environment to create a series of microworlds (White and Horwitz, 1987). This series provides greater control over discovery than is possible with a single microworld. Using a sequence of microworlds, teachers can guide students to slowly add variables and concepts to their view of the natural world until they incorporate both their prior experiences and their classroom experiences into a cohesive, unified model.

The White and Frederiksen (1987) approach also involves instruction in the thinking skills needed to analyze feedback from the microworld. These researchers constructed a curriculum that successfully taught sixth-graders more about force and motion than is typically learned by high school students. The curriculum included (a) a motivational phase which exposed students' misconceptions and inconsistencies in thinking, (b) opportunities to explore key principles governing the behavior of objects, (c) modeling of techniques for making abstract ideas concrete, observable, and qualitative for students, and (d) gradual increases in complexity that allowed students to build on accurate prior knowledge. As students constructed accurate understanding of these scientific phenomenon, Newton's laws of motion were introduced to summarize the observations. The developers reasoned that, ultimately, students needed a philosophically-sound view of the scientific enterprise, including understanding of the laws of motion used by scientists. Thus, White and Horwitz focused on imparting the thinking skills of experts as well as on devising a model similar to models used by experts. They have not yet explored how students make the transition from microworlds to naturally-occurring problems.

White and Frederiksen's (1987) implementation of microworlds for instruction incorporates recent research on how students learn and how teaching can become effective. The sequence of microworlds encourages students to combine difficult representations of the same concepts, and thereby gain better understanding of them. The guided discovery limits the information that

students must process at any one time, thereby increasing the likelihood that understanding will arise. Furthermore, the ThinkerTools environment used for the microworlds has the advantage that it is easily reformulated to illustrate a different set of principles. Both students and teachers might modify the ThinkerTools environment to set up new microworlds for students to explore. As students become more sophisticated, they can add features to the microworld that they wish to explore, and teachers can do the same.

In summary, microworlds can be used to develop expert problem-solving skills. A sequence of microworlds is not sufficient to teach students to think like experts; however, such a sequence offers an opportunity to do so. Furthermore, unlike programming, microworlds can address specific scientific phenomena.

Real-Time Data Collection

Microworlds allow students to explore phenomena not readily accessible to experimentation. In contrast, real-time data collection facilitates and simplifies experiments conducted by students. Using real-time data collection, students can collect, analyze and display information just as expert scientists do. Science teachers can spend less time ensuring that accurate data is collected, and concentrate on expert thinking skills, such as analyzing and interpreting data.

Several effective real-time data collection environments exist. All use probes connected to microcomputers and provide graphic output (see Figure 4). Laws (1986) created opportunities for students to gather information from experiments involving temperature and sound. The Bank Street College of Education included real-time data collection among the activities offered in the Voyage of the Mimi curriculum (Pea & Sheingold, 1987) , and Tinker (1987) at Technical Education Research Centers has devised a vast array of real-time collection techniques for temperature, light, sound, motion, and other scientific phenomena.

As was the case for microworlds, the availability of real-time data collection is far from sufficient for effective science instruction. Curriculum materials emphasizing the problem-solving skills students need make this approach effective. For example, the Computer as Laboratory Partner (CLP) project has implemented a real-time data collection curriculum for thermodynamics

in a local middle school and refined the curriculum using principles from learning and instruction (Linn & Songer, 1988; Linn, Layman, & Nachmias, 1987; Nachmias & Linn, 1987; Friedler, Nachmias, & Linn, 1987; Striley, 1988; Songer & Linn, 1988).

One unanticipated consequence of real-time collection was that students were far more effective at interpreting graphs presented on the computer screen than at interpreting graphs they constructed by hand (Linn, Layman, & Nachmias, 1987; Brasell, 1987; Mokros & Tinker, 1987). Essentially, the dynamic presentation of information in graphic form reinforces for students the idea that a graph represents a relationship rather than a picture. Thus, students can link the temperature graph in Figure 4 to the evaporation of water and alcohol by watching the computer screen while waving their temperature probes. Without MBL experience, students commonly interpret motion graphs as pictures, assuming that when the graph goes up, the object being graphed is going up a hill. In the CLP project, Linn, Layman, and Nachmias (1987) found that students could transfer their understanding of temperature graphs to motion graphs representing the speed of a bicycle over time.

To impart the distinction between heat energy and temperature, the CLP project devised a curriculum, identified a realistic setting, tested four versions of the curriculum, and reformulated the curriculum in light of student performance, after each trial. The researchers changed the cognitive demands placed on students, but not the activities, as they reformulated the curriculum. Enhancements to the cognitive demands, resulted in a fourfold increase in learning outcomes (Linn & Songer, 1988), demonstrating the importance of the process of curriculum reformulation. Several principles governed reformulation. First, the CLP Project sought an appropriate qualitative model for thermodynamics, consistent with a variety of research studies concluding that students need mental models or robust representations of scientific phenomena to understand them effectively (see Gentner & Stevens, 1983; Smith & Goodman, 1984). Many textbooks emphasize the computation of gains or losses in calories or degrees centigrade to represent heat and temperature. The CLP Project found that this quantitative representation of thermodynamics throws a veil of numbers over the distinction between heat energy and temperature. Ultimately, the

researchers chose a qualitative model focused on heat flow and representing heat as a massless entity.

Second, the researchers emphasized the thinking skills of experts, including self-monitoring and self-regulation in reformulating the curriculum. Eventually, students need to learn to direct their own learning, and to be responsible for integrating their own understanding. The CLP curriculum transferred responsibility for complex problem-solving procedures from the teacher to the students. Following research results reported by Brown and Palincsar (1987), the CLP project emphasized having the teacher model self-monitoring skills and then supporting students as they imitated the skills (Linn & Songer, 1988).

The third principle governing reformulation concerned motivation. Students motivated to study science learn more and work more independently. CLP students report great excitement about scientific investigation (Kirkpatrick, 1987). Students like working with technology because the technology responds to them (Lepper, 1985) and report feeling that science is important because they work with computers. Students also recognize that they are responsible for their own learning. One female student, when asked by a reporter to comment on the CLP curriculum remarked, "For the first time I feel like I can find something out for myself in science."

The CLP curriculum allocates 12 weeks to thermodynamics rather than the usual single week in order to teach complex reasoning skills. Allocating extensive instructional time to thermodynamics helped 13-year-olds gain deeper understanding of heat energy and temperature than is typically achieved by 17-year-olds (Songer & Linn, 1988). Furthermore, although students in the Computer as Lab Partner curriculum spent 12 weeks on thermodynamics, they did as well as or better than their peers on standardized tests, no doubt in part because they were more proficient than students in traditional programs at interpreting graphs.

In summary, real-time data collection is a tool used by experts to solve complicated scientific problems. Curriculum reformulation yielded a real-time data collection curriculum that provided robust models of scientific phenomenon, encouraged self-monitoring skills, and motivated students to participate in science. When combined with an effective curriculum, in this case

developed over numerous trials and refinements, it is possible for real-time data collection to greatly enhance students' understanding of complex phenomena and their ability to solve new scientific problems. An additional consequence of real-time data collection is that students gain expert skill in interpreting graphs and can transfer this understanding to new problems. Students are likely to perform well on graphing items on standardized tests, as the result of experience with real-time data collection. Thus, this tool of experts does, at minimum, impart graphing skills needed by science students, in contrast to other expert tools such as programming.

Modeling of Scientific Phenomenon

Experts frequently use technological tools to model scientific phenomenon such as the lifecycle of stars, the interactions of the heart and lung system, and the spread of disease. Scientists create models based on their theoretical ideas and then test them against experimental data. Recently-developed software makes this tool available to students. The STELLA (Structural Thinking, Experimental Learning, Laboratory with Animation; High-Performance Systems, Inc., 1985) software allows students to define and test complex models without understanding the underlying mathematics.

Using STELLA, Mandinach (1987, 1988) and her colleagues at the Educational Testing Service have taught high school biology, chemistry, physics, physical science, and history teachers to use STELLA and then helped them design activities to take advantage of the software in their classes. They found that students designed complicated models and refined their models in light of empirical data, used models designed by others to gain understanding of scientific principles, and learned the value of computer modeling (Mandinach & Thorpe, 1987a,b).

As for other tools of experts, a curriculum to accompany STELLA is needed to teach the skills experts use to model scientific phenomena. To design a model, experts design a problem solution, specify the factors likely to influence the outcome, select appropriate values and ranges for these factors, and then indicate how these factors might interact in the experimental situation. Once a model has been designed and analyzed, experts test the effectiveness of the model and reformulate the model when tests reveal deficiencies. Thus, engaging in the modeling of scientific

phenomenon and the refinement of the model in light of empirical data offers opportunities for development of important thinking skills. Mandinach (1988) is working with instructors to develop teacher roles and curriculum materials that impart these skills.

Intelligent Tutoring

The success of expert systems for well defined tasks, such as configuration of a computer system or analysis of a geological sample from a drill hole has motivated developers to determine whether students could learn experts' techniques through the use of intelligent tutors. Just as expert scientists have been replaced by expert systems in some tasks, the argument is that expert teachers might be replaced by intelligent tutors for some domains.

However, emulating the behavior of expert scientists on narrow problems is considerably easier than emulating the behavior of expert teachers. In addition, those designing intelligent tutors have looked to learning theory more than to the behavior of expert teachers in implementing designs. Anderson and his colleagues (Anderson, Boyle, & Reiser, 1985) have designed intelligent tutors governed by the ACT theory of learning (Anderson, 1976). These tutors teach domains with a small set of rules governing performance, such as algebra symbol manipulation, geometry proof construction, and Lisp computer programming. These tutors address the traditional goals for instruction found in textbooks. Thus, the geometry tutor teaches the rules of geometry. In algebra, the tutor teaches students to manipulate algebraic symbols, and the Lisp tutor teaches students to write Lisp functions.

Based on ACT theory, intelligent tutors developed by Anderson and his group at Carnegie-Mellon University create a model of how the student solves problems, provide feedback to guide the student to imitate the correct model, and select appropriate problems based on prior performance of the student. The tutor has difficulty interpreting responses when the student strays from an expected solution path, tutors quickly correct students if they appear to be heading down a blind alley, but accept many valid solutions to problems.

Evaluation suggests that Anderson's intelligent tutors are reasonably successful in imparting the information they are designed to teach, but they do not emphasize the problem-solving skills of

experts. Thus, the algebra tutor has been successful in teaching algebraic manipulation, but less successful when the emphasis is on designing problem solutions. The geometry tutor is not always consistent with classroom goals because some teachers place little emphasis on proofs and others resist the representation of the geometry proof used in the tutor. Although the representation offered by the geometry tutor is at least as effective as the traditional two-column proof, when students are required to produce two-column proofs, instruction using a different method requires that students recognize the similarities and reformulate their experiences in the two-column proof format.

Other evaluations suggest that students learn the strategies the tutor teaches but are not convinced that these strategies constitute an exhaustive set (Reiser, 1988). Thus, students comment that although they finally solved the problem the way the tutor insisted, they believe their solution to the problem would also succeed. Providing a tutor with sufficient information to explain why alternative solutions are unsuccessful would seem to be an interesting goal for future intelligent tutors.

Thus, current intelligent tutors focus on the content goals of science or mathematics textbooks rather than on the thinking skills of experts. Furthermore, these tutors are limited to rule-governed domains. Finally, it seems clear that expert teachers perform functions not represented in theories governing current intelligent tutors, including empathizing with the student, selectively providing feedback and rarely indicating that students are incorrect (Lepper & Chabay, 1985). The technology used for the design of expert systems provides direct feedback to learners but does not provide the thinking skills emphasized by expert teachers, including (a) encouraging students to proceed until they recognize their own errors or (b) teaching students to recognize when they need feedback. Nevertheless, intelligent tutors provide an ideal laboratory for investigating the effectiveness of the instructional theories used to design them. Tutors can be programmed to contrast one instructional approach with another and can reliably implement alternative instructional approaches.

Databases

Another technological tool used by experts is the electronic database. Visionaries suggest that electronic databases will replace almanacs, taxonomies, and textbooks as a source of up-to-date information about a vast array of things. Hypermedia environments that allow pictures, graphs, text, and other information to be linked in non-linear patterns offer considerable promise (Goodman, 1987; Nelson, 1987). The availability of HyperCard (Atkinson, 1987) and similar products may increase effective use of databases for precollege instruction. As with other technological tools used by experts, however, techniques for searching, accessing, and utilizing database information remain to be clearly delineated or effectively taught.

For example, databases are particularly important in medical school instruction (Olivieri, 1987). Medical school instructors are converting complex lectures to hypermedia stacks. Thus, a lecture on pneumonia might include videodisc presented slides of viruses and bacteria that cause pneumonia, as well as videodisc presented images of x-rays of individuals suffering from various forms of pneumonia. All this information could be linked to a core set of principles about pneumonia, and the principles could also point to information about appropriate treatments, including different forms of antibiotics, as well as medical histories of particular patients with puzzling symptoms. Ultimately, such databases might be presented in the context of instruction in medical school, sold to the student as a study guide, and updated regularly. The student would then use this database throughout her medical career to look up information when presented with complex diagnostic problems. Current databases available for precollege science instruction are less complex and modifiable than the medical school materials.

Databases thus have the possibility of encouraging students to integrate information and engage in self-monitoring. For example, databases may provide students with models of appropriate relationships between information. Whereas textbooks, by presenting information in a linear, sequential form, may discourage students from integrating it, databases can illustrate the rich relationships among information as well as systematic patterns of relationships. Hypermedia environments can go further to provide examples of the networking of ideas. Databases also provide feedback to students when they investigate hypotheses. For example, if students had a

database with information about animals, they could investigate hypotheses such as the relationship between heart rate and animal size. To make these tools effective, teachers and curriculum developers need to provide models of integrated information and feedback on student progress.

Summary

In summary, the tools of experts used in science classes in Stage II provide the opportunity to use complex problem-solving skills, but are not sufficient to elicit or impart the problem-solving strategies that experts use. In fact, as programming instruction has illustrated, using the tools of experts might require novices to engage in memorizing syntax rather than problem solving. To make effective use of the tools of experts, teachers and curriculum developers have looked to research on learning and instruction for guidance and have begun to identify ways to impart the problem-solving skills that experts use.

Trial and refinement. Finding effective ways to use principles from learning and instruction in conjunction with the tools of experts requires collaborative trial and refinement because so many complex factors interact in the classroom. All the projects incorporating the tools of experts have benefitted from trying the materials in somewhat realistic settings involving teachers, researchers, administrators, and policy makers, analyzing the effectiveness of the materials in realistic settings, and refining the curriculum materials in order to make the instruction more effective. The most dramatic evidence for curriculum reformulation was found for the CLP curriculum, where changing the cognitive demands of the curriculum, but not the activities used in the curriculum resulted in a fourfold increase in student understanding of principles of thermodynamics.

Teaching thinking. Introducing the tools of experts into the science curriculum leads to changes in the goals of science instruction and the roles of science teachers. As knowledge proliferates, students and citizens alike will need the problem-solving and self-monitoring skills used by scientists. For example, with the availability of modeling tools, students will have the opportunity to create and test ideas about complex phenomena, whereas lacking problem-solving and self-monitoring skills, these same students will be unable to use modeling software effectively. Thus, expert tools facilitate modification of the goals of science curricula and the roles of science

teachers, moving away from emphasis on absorbing scientific knowledge and towards emphasis on designing, predicting, analyzing, and interpreting scientific events.

As discussed above, to engage in problem solving and self-regulation, students must pursue a single topic for more instructional time than is required to memorize information. Fleeting coverage often introduces more vocabulary words in each week of science instruction than are introduced in foreign language classes at the same school. In order to teach reasoning, it will be impossible to cover as many topics as are traditionally included in precollege science courses. Nevertheless, students need to know about the subject matter in order to design problem solutions and monitor their own problem-solving strategies. They need teachers to model problem solving, guide student behavior, and encourage self-monitoring.

Technology. In Stage II, more powerful hardware and software allowed developers to use graphics, real-time data collections, and large databases. It was possible, using workstations, to build intelligent tutors. In addition, the installed base of technology in education expanded. Nevertheless, the school market remained marginal -- few schools budgeted for maintenance, much less software. As a result, most technological developments devised at schools were supported by government grants. Using the tools of experts was often the only software alternative and even for programming, few curriculum materials existed.

During this stage, Logo emerged as a possible courseware authoring environment, several computer operating systems were enhanced to include "tool boxes" that made development easier, and software designers began to realize the potential advantages of creating specialized tools for those using technology in instruction.

Stage III: Integrating Technology and Learning

At the onset, we argued that technology shapes instruction, that educators shape the development of new technological tools, and that ultimately, curriculum developers, hardware and software developers, researchers, and those involved in realistic settings will interact synergistically to achieve reformulated goals of science education.

Attempts to achieve this third stage of understanding are constrained by currently available educational settings. Many have proposed reformulations of instructional delivery to incorporate technological advance (Shanker, 1987a,b; Futrell, 1986, 1987; Linn, 1987a,b). However, these reformulations involve political, economic, and sociological forces that are beyond the scope of this paper.

The other major issues in Stage III concern (a) synergistic combination of technological innovation with increased understanding of learning and instruction, and (b) the development of flexible, reformulatable technological tools better suited to the process of curriculum reformulation than the tools of experts characteristic of Stage II.

Technology and Research on Learning and Instruction

Many opportunities for synergistically combining technological advance with principles from learning and instruction are becoming apparent. Examples for cooperative problem solving, expert knowledge representation, and instructional flexibility illustrate this trend.

Cooperative problem solving. One example concerns teaching students collaborative problem solving. It is clear that complex scientific problems require a community of scholars and that joint problem solving skill is needed by effective researchers. Combining technological advance with analysis of how collaborative problem solving might be taught could lead to effective ways to teach collaborative problem solving in science classes.

Some preliminary insights come from the CLP classroom where the instructor encouraged students to collaborate. Striley (1988) noted that the 16 monitors present in the classroom facilitated collaboration. When anomalous results appeared on a monitor, groups of students would join those conducting the experiment and attempt to jointly resolve the anomaly. Thus, the presence of feedback encouraged students to resolve discrepancies, facilitated joint interest in scientific problems, and encouraged discussion.

Other insights came from efforts to teach students to solve complex, naturally occurring problems. Often science curricula ignore large problems which cannot be addressed by individual students, even though these problems could be effective for teaching complex problem-solving

skills. Students working jointly can divide larger, more complex problems into components, work their own piece of the problem, and then report back to the group. This model, characteristic of scientific research groups, seems equally amenable to science instruction. Technological tools can help groups of students integrate their efforts at problem solving through electronic mail and feedback on partial solutions. For example, in Pascal programming, groups can work on large, complex problems that they could not solve individually. Each group can evaluate their part of the solution individually before the pieces are combined. This innovation has the desired additional advantage that it forces students to proceduralize their code so the code of others will not interact with it.

Kidnet, discussed elsewhere in this volume, is another example of the advantages of collaborative problem solving with technology (Tinker, 1987). Using electronic mail, students across the country can create databases of the results from research investigating the same question and review these systematically. Linn and Clancy (1988) found another way to help students learn about large complex problems. They have students practice collaborative skills by modifying programs written by experts. Thus, students are asked to reformulate a portion of a complex problem solution, to debug a program, or test a solution. The technological environment provides students with an opportunity to analyze solutions to problems they could not generate themselves, and to generate a problem solution that they could contribute to but not individually devise. Furthermore, these researchers provide experts commentary to impart the thinking skills needed for the task.

To make these efforts effective, teachers model cooperative problem solving and support students when they use the techniques. Furthermore, the curriculum emphasizes and rewards group problem-solving skills rather than only individual processes. Progress will require that curriculum developers, teachers, and software developers work together to create materials such as these.

Expert knowledge organization. Another opportunity for combining technology and curriculum development concerns imparting the knowledge organization used by experts. Expert

problem solvers learn generalizable algorithms, templates, or plans that they can apply to many similar problems. These algorithms or templates are often not apparent from inspecting the expert's solution, and are often not emphasized in technological tools used by experts, although they may be implicit. As Soloway (1988) suggests, the templates that experts use can be stored in a database and made more explicit by providing shortcuts for applying common templates in model-building programming. Effectiveness is increased by providing solutions that illustrate how these templates are used by experts and by having teachers model the use of the templates. Thus, by reformulating technological tools as well as curriculum materials, it is possible to help students gain understanding of templates or algorithms for problem solutions, and recognize the advantages of looking for reusable chunks of information rather than learning isolated pieces of information. Such an emphasis allows teachers to encourage students to reflect on their own problem-solving approaches and become better at self-monitoring.

Instructional flexibility. A third opportunity arising from a synergy between technological advance, curriculum innovation, and increased understanding of the learner concerns instructional flexibility. With on-line curriculum materials, large databases of software, images and catalogs, and electronic mail, teachers could tailor their instruction to their own needs and interests and those of their students. For example, a teacher could choose to provide deep coverage of motion one year and of magnetism the next. In each case, appropriate written materials could be selected from the database and printed for students, appropriate software could be ordered electronically, and appropriate activities could be requested from a database of materials in the central storage area. Rather than relying on rapidly outmoded textbooks, teachers would draw on recently written materials. Rather than omitting images, software, and experiments because they are difficult to locate, these would be retrieved from CD-ROM storage, databases of real-time data collection experiments or simulations, and centrally stored equipment. Finally, in this futuristic environment, the "standardized" test items assigned to students would reflect the topics covered by the teacher and would be administered interactively. These resources would not, by themselves, impart the thinking skills and collaborative learning skills students need, but they would permit teachers to

focus on complex problem solving and permit students to explore interesting and timely topics. Eventually, such curricular resources could be used by students to design personalized curricula or conduct sustained investigations of scientific phenomena.

Technological Tools of the Future

Each of the suggestions for combining advances in technology with advances in learning and instruction suggest reformulation of the technological environment the student uses. One of the drawbacks of implementing the tools of experts in the classroom is that often those tools are quite unmodifiable and not designed for instruction. Clearly, greater progress in Stage III development would result if those developing the materials could rapidly modify the technological environment as well as the curriculum.

Recently, authoring environments of several sorts have emerged to alleviate this situation. First, more powerful courseware authoring environments are now available. Computer Tutor, developed by Sherwood and Sherwood (1986) at Carnegie-Mellon University, allows non-professional programmers to devise curriculum materials. Languages such as True Basic and Light Speed Pascal access the Macintosh toolbox and permit much more rapid development of instructional materials than was possible in the past. Using the HyperCard environment, it is possible to launch other applications such as Light Speed Pascal or STELLA, thereby permitting developers to design an interface between the student and the application, rather than attempting to augment the application. Another new authoring system, Course of Action (Authorware, 1987), also integrates well with tools used by experts, allowing developers to incorporate more pedagogical principles into technological materials. In addition, authoring environments such as Course of Action and HyperCard, combined with expert tools such as Light Speed Pascal or STELLA, allow easy reformulation of the instructional component of the curriculum, yet incorporate a complex, expert tool that would be beyond the development capabilities of most curriculum developers.

Conclusions

A great deal of progress has been made in combining technological advance with insights into science learning and science instruction to improve science education. At first, technological tools were used to achieve the established goals of science education. In the second stage, the tools of experts formed a good starting point for effective use of technology in science classrooms, and make it possible to focus on the thinking skills used by experts. The third stage forms the beginning of efforts to target science education to the complex needs of society. This process involves redefinition of the roles of teachers, technology, textbooks, experiments, and other influences to evolve the science programs that will restore our nation to prominence in education. To accomplish this, priorities include:

First, courseware authoring environments that facilitate reformulation of educational software will improve science instruction. Such environments make it possible to tailor curriculum materials to teachers and students, and greatly reduce the cost of development.

Second, taking full advantage of the talents of teachers in modeling scientific processes, encouraging students to learn, and helping learners gain coherent understanding will improve science education. To do so requires improved teacher preparation, enhanced teacher status and a reconceptualization of school organization, as well as better understanding of how teachers can use technology.

Third, substantial trial and refinement of innovations in realistic settings will help them reach their full potential. This process has succeeded in the CLP project, the ThinkerTools project, and others discussed above. In the past, the Science Curriculum Improvement Study (Karplus, 1975) and the Health Activities Program both benefitted from trial and refinement. In the area of reading instruction, the KEEP Program in Hawaii benefitted from this process (Vogt, Jordan, & Tharp, 1987; Tharp, et al., 1984), as did efforts to design reciprocal teaching (Brown & Palincsar, 1987).

Fourth, to impart the thinking and reasoning skills that students need in the information age, the goals and emphasis of science education require reformulation. Technological advances, such as the availability of on-line tailorable science curricula, reduce reliance on textbooks and facilitate

changes in the goals. Simulations, microworlds, and on-line feedback make emphasis on the process of science more practical.

Fifth, effective technological innovation in science education depends on true collaboration among curriculum developers, precollege teachers, educational researchers, school administrators, and state and national policy makers. These individuals must work with motivated students and supportive parents in order to fully achieve the science instruction needed in our schools. These collaborations are necessary to guide the trial and refinement that leads to effective instruction. As a result, many national groups have called for funding of Centers for Collaboration in Science Education or similar entities (e.g., Linn, 1987b; Pea & Soloway, 1988).

Let us jointly address these important problems and move to impart more cohesive, robust, self-regulated understanding of science. A philosophically-sound view of science can arise if students spend time engaged in guided discovery, emulating the problem-solving skills of the teacher and using feedback from the technological environment.

Bibliography

Anderson, J. R. (1976). Language, memory, and thought. Hillsdale, NJ: Lawrence Erlbaum Associates.

Arons, A., Bork, A. M., Franklin, S., Kurtz, B., & Collea, F. (1981). Science literacy in the public library -- Batteries and bulbs. In *Proceedings of the National Educational Computing Conference*, June, 1981.

Anderson, J. R., Boyle, C. F., & Reiser, B. J. (1985). Intelligent tutoring systems. *Science, 228,* 456-467.

Atkinson, B. (1987). *HyperCard* [Computer program]. Cupertino, CA: Apple Computer, Inc.

Authorware, Inc. (1987). *Course of action*™ [Computer program]. Bloomington, MN: Author.

Becker, H. J. (1986). Our national report card: Preliminary results from the New Johns Hopkins Survey. *Classroom Computer Learning, 2,* 30-33.

Bloom, B. S. (1984). The 2 sigma problem: The search for methods of group instruction as effective as one-to-one tutoring. *Educational Researcher, 13,* 4-16.

Bork, A. M. (1980). *Computer-assisted learning physics education.* New York: Pergamon Press.

Bork, A. M. (1981). *Learning with computers.* Bedford, MA: Digital Press.

Brasell, H. (1987). The effect of real-time laboratory graphing on learning graphic representations of distance and velocity. *Journal of Research in Science Teaching,* 24(4), 385-95.

Broad, W. J. (1988, March 7). Novel technique shows Japanese outpace Americans in innovation. *The New York Times,* 1.

Brown, A. L. & Palincsar, A. S. (1987). Reciprocal teaching of comprehension strategies: A natural history of one program for enhancing learning. In J. G. Borkowski & J. D. Day (Eds.), *Cognition in special children: Comparative approaches to retardation, learning disabilities, and giftedness.* Norwood, NJ: Ablex Publishing.

Clement, J. (1987, April). *Observed methods for generating analogies in scientific problem solving.* Paper presented at the annual meeting of the American Educational Research Association, Washington, D. C.

di Sessa, A. (1979). Learning physics with dynaturtle. In S. Papert, D. Watt, A. di Sessa, & S. Weir (Eds.), *Final report of the Brookline Logo project, Part II: Project summary and data analysis* (Memo 545, pp. 6.1 - 6.20). Cambridge, MA: MIT, AI Laboratory.

di Sessa, A. (1986). Artificial worlds and real experience. *Instructional Science, 14*(3/4), 207-227.

Friedler, Y., Nachmias, R., & Linn, M. C. (1987, April). Using microcomputer-based laboratories to foster scientific reasoning skills. In M.C. Linn (Chair) *Microcomputer-based*

laboratories: Theory into practice. Symposium conducted at the annual meeting of the National Association for Research in Science Teaching, Washington, D.C.

Futrell, M. H. (1986). Restructuring teaching: A call for research. *Educational Researcher,* 15, 5-8.

Futrell, M. H. (1987). The Holmes Group report: A teacher's perspective. *Teachers College Record, 88,* 378-383.

Gentner, D. & Stevens, A. L. (Eds.). (1983). *Mental models.* Hillsdale, NJ: Lawrence Erlbaum Associates.

Goodman, D. (1987). *The complete HyperCard handbook.* New York: Bantam Books.

High-Performance Systems, Inc. (1985). *STELLA* [Computer program]. Hanover, NH: Author.

Hughes, M. & Macleod, H. (1986). Using Logo with very young children. In R. W. Lawler, B. du Boulay, M. Hughes, & H. Macleod, *Cognition and Computers* (pp. 179-219). West Sussex: Ellis Horwood Limited.

International Association for the Evaluation of Educational Achievement (IEA). (1988). *Science achievement in seventeen countries: A preliminary report.* New York: Pergamon Press.

Karplus, R. (1975). Strategies in curriculum development: The SCIS Project. In J. Schaffarzick & D. H. Hampson (Eds.), *Strategies for curriculum development.* Berkeley, CA: McCutchan Publishing Corp.

Kirkpatrick, D. (1987). Technology in education: A middle school teacher's experience. *Technology and Learning, 1*(3), 4-5. OK

Lawler, R. W. (1986). Natural learning: People, computers and everyday number knowledge. In R. W. Lawler, B. du Boulay, M. Hughes, & H. Macleod, *Cognition and Computers* (pp. 9-80). West Sussex: Ellis Horwood Limited.

Laws, P. W. (1986, June). *Workshop physics* (Proposal to the Fund for the Improvement of Post Secondary Education, U.S. Department of Education). Carlisle, PA: Dickinson College, Department of Physics and Astronomy.

Lepper, M. R. & Chabay, R. W. (1985). Intrinsic motivation and instruction: Conflicting views on the role of motivational processes in computer-based education. *Educational Psychologist, 20,* 217-230.

Linn, M. C. (1985). The cognitive consequences of programming instruction in classrooms. *Educational Researcher, 14,* 14-16, 25-29.

Linn, M. C. (1987a). Establishing a research base for science education: Challenges, trends and recommendations. *Journal of Research in Science Teaching, 24*(3) 191-216.

Linn, M. C. (1987b). *Education and the challenge of technology.* Cupertino, CA: Apple Computer, Inc. (Proceedings of a Conference on Technology and Education, August, 1986).

Linn, M. C. & Clancy, M. (January, 1988). *Programming case studies: Effect on instruction.* Unpublished manuscript, University of California, ACCEL Project, Berkeley.

Linn, M. C., Layman, J., & Nachmias, R. (1987). The cognitive consequences of microcomputer-based laboratories: Graphing skills development. *Journal of Contemporary Educational Psychology, 12*(3), 244-253.

Linn, M. C. & Songer, N. B. (1988). *Cognitive research and instruction: Incorporating technology into the science curriculum.* Unpublished manuscript, University of California, Computer as Lab Partner Project, Berkeley.

Mandinach, E. B. (1988). *The cognitive effects of simulation-modeling software and systems thinking on learning and achievement.* Princeton, NJ: Educational Testing Service.

Mandinach, E. B. & Thorpe, M. E. (1987a). Caveats and realities in technological curriculum innovation. *Technology and Learning, 1*(4), 1-3, 7, 9.

Mandinach, E. B. & Thorpe, M. E. (1987b). The systems thinking and curriculum innovation project. *Technology and Learning, 1*(2), 1, 10-11.

Mandinach, E. B. & Linn, M. C. (1987). Cognitive consequences of programming: Achievements of experienced and talented programmers. *Journal of Educational Computing Research, 3*(1), 53-72.

Mokros, J. R., & Tinker, R. F. (1987). The impact of microcomputer-based labs on children's ability to interpret graphs. *Journal of Research in Science Teaching, 24*(5), 369-383.

Nachmias, R., Friedler, Y., & Linn, M. C. (1988). *The role of programming environments in Pascal instruction.* Unpublished manuscript, University of California, Computers and Problem Solving Project, Berkeley.

Nachmias, R. & Linn, M. C. (1987). Evaluations of science laboratory data: The role of computer-presented information. *Journal of Research in Science Teaching, 24*(5), 491-506.

National Assessment of Educational Progress (NAEP). (1978). *Three national assessments of science: Changes in achievement 1969-77* (Report No. 08-S-00). Denver: Education Commission of the States.

National Commission on Excellence in Education. (1983). *A nation at risk: The imperative for educational reform.* Washington, DC: U.S. Government Printing Office.

National Science Foundation. (1987, December). *International science and technology data update.* Washington, D.C.: Author.

National Science Board. (1988). *Science and engineering indicators -- 1987.* Wahington, D.C.: National Science Foundation.

Nelson, T. H. (1987). *Computer lib* (2nd ed.). Redmond, WA: Tempus Books (Microsoft Press).

Olivieri, P. (Ed.). (1987). A HyperCard application: Med-videodisc manager. In *Wheels for the mind* (Vol 3, No. 4, An Apple University Consortium publication, pp. 19-20). Chestnut Hill, MA: Boston College and Apple Computer, Inc.

Papert, S. A. (1980). *Mindstorms: Children, computers, and powerful ideas.* New York: Basic Books.

Pea, R. D., & Kurland, M. D. (1987). On the cognitive effects of learning computer programming. In R. D. Pea & K. Sheingold (Eds.), *Mirrors of minds: Patterns of experience in educational computing* (pp. 147-177). Norwood, NJ: Ablex Publishing.

Pea, R. D. & Sheingold, K. (1987). *Mirrors of minds: Patterns of experience in educational computing*. Norwood, NJ: Ablex Publishing.

Pea, R. D. & Soloway, E. (1988). *The state of the art in educational technology R&D: Issues and opportunities*. Report to the Office of Technology Assessment, Washington, D. C.

Reif, F. (1987). Instructional design, cognition, and technology: Applications to the teaching of scientific concepts. *Journal of Research in Science Teaching, 24*(4), 309-324.

Reiser, (1988, March). *Computer tutoring project.* Paper presented at the Army Research Institute meeting, Champaign, Illinois.

Rousseau, J. J. (1892). *Treatise on education* (W. H. Payne, Trans.). New York: Appleton and Company. (Original work published 1793)

Shanker, A. (1987a). Tomorrow's teachers. *Teachers College Record, 88,* 423-429.

Shanker, A. (1987b). A comment on restructuring schooling. *The Journal of Economic Education, 18,* 66-67.

Sherwood, B. A. & Sherwood, J. N. (1986). *The CMU tutor language*. Champaign, IL: Stipes Publishing Co.

Sloane, K. & Linn, M. C. (in press). Instructional conditions in Pascal programming classes. In R. Mayer (Ed.), *Teaching and learning computer programming: Multiple research perspectives*. Hillsdale, N.J.: Lawrence Erlbaum Associates.

Smith, E. E. & Goodman, L. (1984). Understanding instructions: The role of an explanatory schema. *Cognition and Instruction, 1,* 359-396.

Smith, S. G. & Sherwood, B. A. (1976). Educational uses of the PLATO computer system. *Science, 192,* 334-352.

Soloway, E. (1988, April). *Tools vs. assistants: Shedding a poor metaphor for a more illuminating one*. Paper presented at the annual meeting of the American Educational Research Association, New Orleans, LA.

Songer, N. B. & Linn, M. C. (1988). *Everyday problem solving in thermodynamics.* Unpublished manuscript, University of California, Computer as Lab Partner Project, Berkeley.

Stein, J. S. & Linn, M. C. (1985). Capitalizing on computer-based feedback: An investigation of Rocky's Boots. In M. Chen & W. Paisley (Eds.), *Children and microcomputers: Formative studies* (pp. 213-227). Beverly Hills: Sage Publications.

Striley, J. S. (1988). *How students negotiate understanding: Characterizing the problem-solving processes of dyads during microcomputer-based science labs.* Unpublished dissertation, University of California, Berkeley.

Tharp, R. G., Jordan, C. Speidel, G. E., Au, K. H., Klein, T. W., Calkins, R. P., Sloat, K. C. M., & Gallimore, R. (1984). *Product and process in applied developmental research: Education and the children of a minority*. Honolulu, HI: The Kamehameha Schools.

Tinker, R. F. (1987). Network science arrives. National Geographic-coordinated school experiment generates student-scientists. *Hands On!, 10*(1), 1, 10-11. (A publication of Technical Education Research Centers, Cambridge, MA)

Turkle, S. (1984). *The second self: Computers and the human spirit*. New York: Simon & Schuster, Inc.

Vogt, L. A., Jordan, C., & Tharp, R. G. (1987). Explaining school failure, producing school success: Two cases. *Anthropology & Education Quarterly, 18*, 276-382.

Von Blum, R. & Hursh, T. M. (1987). *Tribbles* [Computer program]. Iowa City, IA: Conduit.

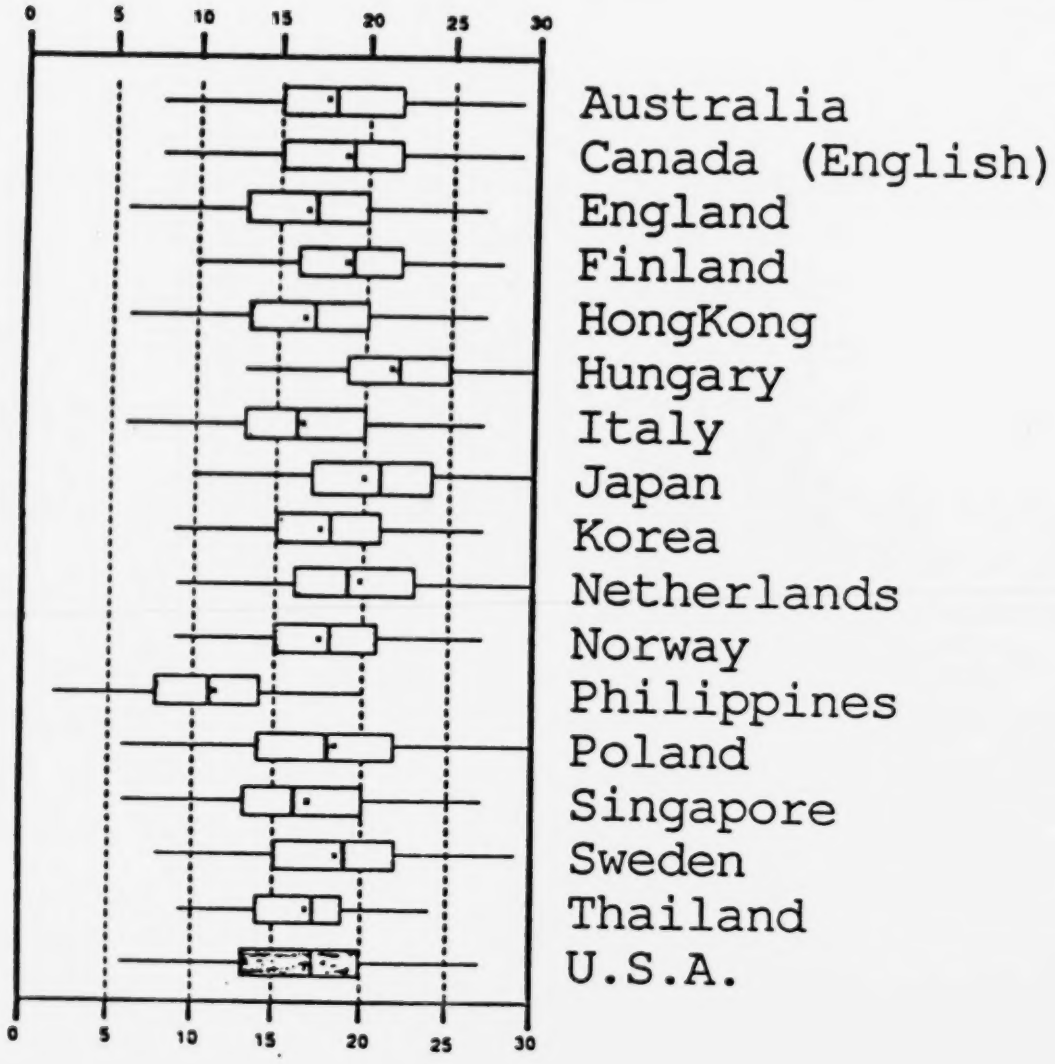
White, B. Y. (1981). *Designing computer games to facilitate learning* (Technical Report No. AI-TR-619). Cambridge, MA: Artificial Intelligence Laboratory.

White, B. Y. (1984). Designing computer activities to help physics students understand Newton's laws of motion. *Cognition and Instruction, 1*, 69-108.

White, B. & Horwitz, P. (1987). *ThinkerTools: Enabling children to understand physical laws* (Report No. 6470). Cambridge, MA: BBN Laboratories Incorporated.

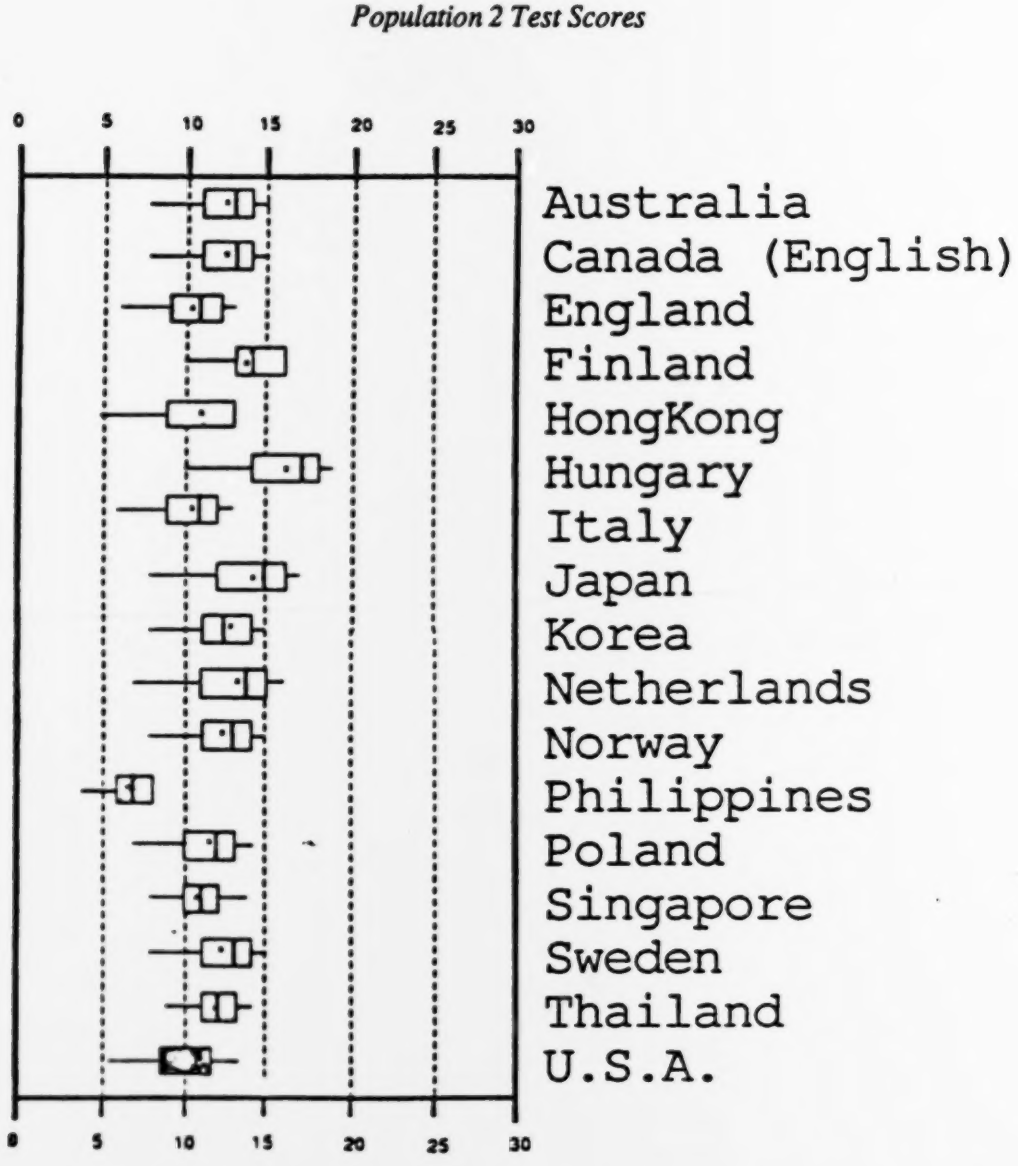
White, B. & Frederiksen, J. R. (1987). *Causal model progressions as a foundation for intelligent learning environments* (Report No. 6686). Cambridge, MA: BBN Laboratories Incorporated.

Figure 1: Core test scores for seventeen countries



Source: International Association for the Evaluation of Educational Achievement (IEA), Science achievement in seventeen countries: A preliminary report.

Figure 2: Core Test Science Scores of the Bottom 25 Percent of Those in School at Population 2 Level



Source: International Association for the Evaluation of Educational Achievement (IEA), Science achievement in seventeen countries: A preliminary report.

Figure 3: A screen display from Macintosh Pascal demonstrating some programming tools and the juxtaposition of the program and its output

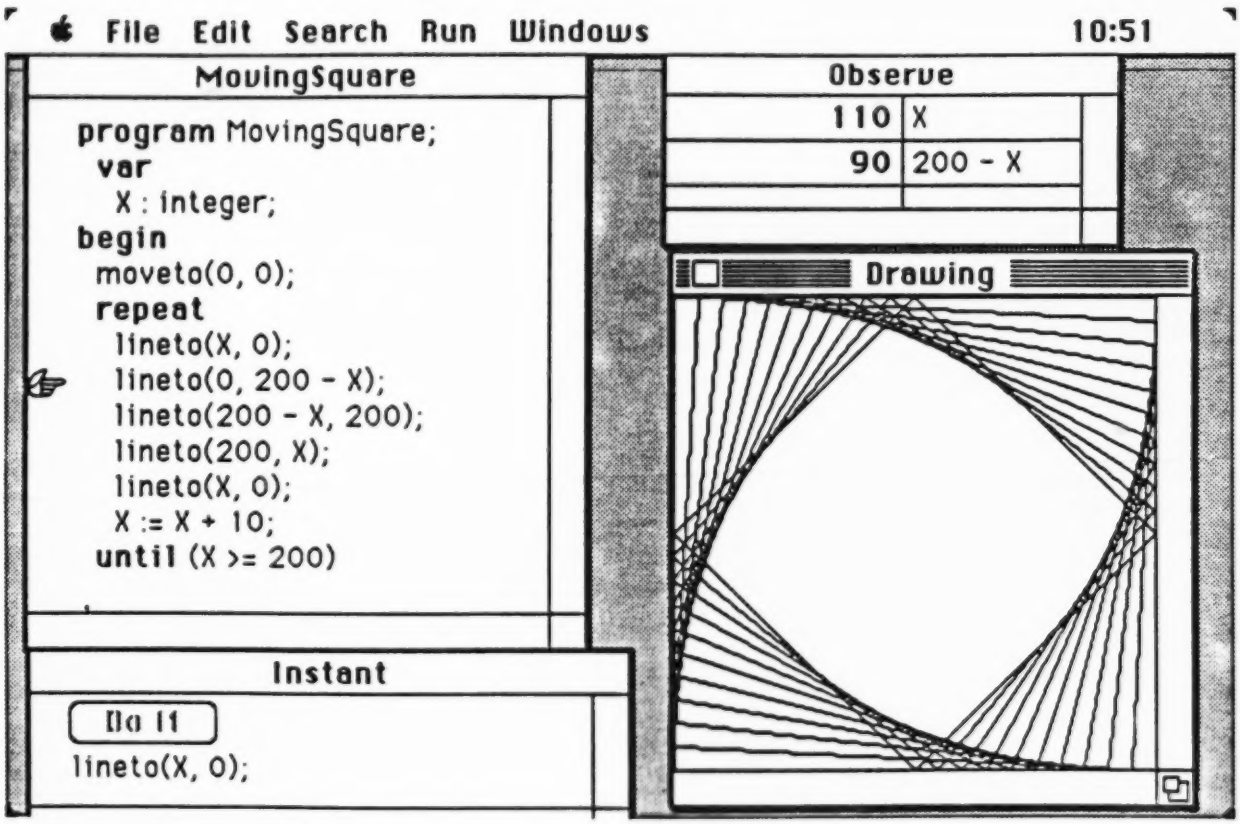


Figure 4: Evaporation graph of alcohol and water